

---

# **Design Document**

## **for**

# **Automated Lecture Hall Booking Portal**

**Version 0.1**

**Prepared by-**

**Group 12**

**Group Name: Aviators**

#	Name	Roll No.	Email
•	Chaitanya Bramhapurikar	230305	bcvishwas23@iitk.ac.in
•	Rahul Meena	230832	rahulkcm23@iitk.ac.in
•	Bhavya Chauhan	230294	bhavyach23@iitk.ac.in
•	Atharv Phirke	230238	atharvp23@iitk.ac.in
•	Aaradhyा Rohi	230011	aaradhyaa23@iitk.ac.in
•	Chaudhari Divyesh	230325	divyesh23@iitk.ac.in
•	Devansh A Dhok	230354	devanshad23@iitk.ac.in
•	Areen Mahich	230188	areenm23@iitk.ac.in
•	Daksh Dua	220321	dakshdua22@iitk.ac.in
•	Avantika Rohite	230245	avantikar23@iitk.ac.in

**Course:** CS253  
**Mentor TA:** Souvik Mukherjee  
**Date:** 7 February 2025

# Contents

<b>1 Context Design</b>	<b>4</b>
1.1 Context Model . . . . .	4
1.1.1 Professors, Clubs, and Societies . . . . .	4
1.1.2 LHC Booking Portal . . . . .	4
1.1.3 Authorities (General Secretary, DOAA, Faculty Advisors) . . . . .	4
1.1.4 Database . . . . .	5
1.1.5 Email API . . . . .	5
1.2 Human Interface Design . . . . .	6
<b>2 Architecture Design</b>	<b>10</b>
<b>3 Object Oriented Design</b>	<b>11</b>
3.1 Use Case Diagram . . . . .	11
3.2 Class Diagram . . . . .	12
3.3 Sequence Diagram . . . . .	13
3.4 State Diagram . . . . .	15
<b>4 Project Plan</b>	<b>16</b>
<b>Appendix A: Group Log</b>	<b>18</b>

## **Revisions**

<b>Version</b>	<b>Primary Author(s)</b>	<b>Description of Version</b>	<b>Date Completed</b>
0.1	Group 12	Initial Draft	7/02/25

# 1 Context Design

## 1.1 Context Model

The context model defines the scope and boundaries of the LHC Booking Portal and its interactions with external entities, such as professors, clubs and societies, LHC staff, and external systems like an email API and a database. The system is designed to streamline the booking process while ensuring proper validation, approval, and tracking of bookings. Below are the key components and their interactions:

### 1.1.1 Professors, Clubs, and Societies

- **Interaction:** These entities act as the primary users who can submit booking requests to reserve lecture halls or common rooms.
- **Capabilities:**
  - Submit booking requests with details such as date, time, and purpose.
  - Check available time slots and download the reservation history.
  - Receive email confirmations for approved or rejected requests.

### 1.1.2 LHC Booking Portal

- **Core Functionality:**
  - Acts as the central system to process booking requests.
  - Manages interactions with other entities such as the database, email API, and authorities.
- **Workflow:**
  - Receives requests from users and forwards them to the Authorities for approval.
  - Interacts with the Database to fetch or update information such as room availability and booking history.
  - Uses the Email API to send notifications regarding booking status.

### 1.1.3 Authorities (General Secretary, DOAA, Faculty Advisors)

- **Role:**
  - Responsible for approving or rejecting booking requests based on institutional policies.
- **Interaction:**
  - Receive authentication emails generated by the portal for action.
  - Communicate their decision back to the LHC Booking Portal for further processing.

#### 1.1.4 Database

- **Purpose:**

- Stores all relevant data, including user profiles, booking requests, room availability, and historical records.

- **Interaction:**

- Provides data to the portal for checking room availability and validating requests.
- Updates the database with the latest booking information and user feedback.

#### 1.1.5 Email API

- **Functionality:**

- Automates the generation and sending of emails for various purposes, such as:
  - Authentication requests to authorities.
  - Booking confirmations or rejections to users.
- Ensures all communications are prompt and well-documented.

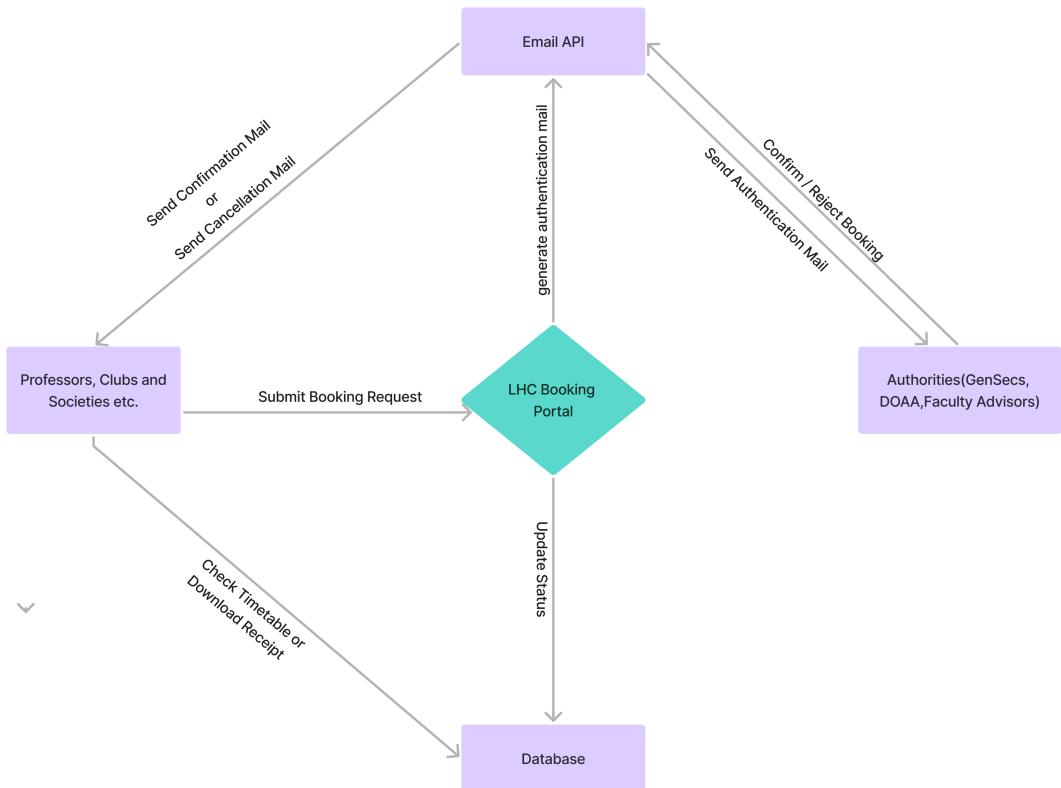


Figure 1: Context Model

## 1.2 Human Interface Design

```
Login Page
verifycredentials(username, password);
isADMIN(username);
isUSER(username);
```

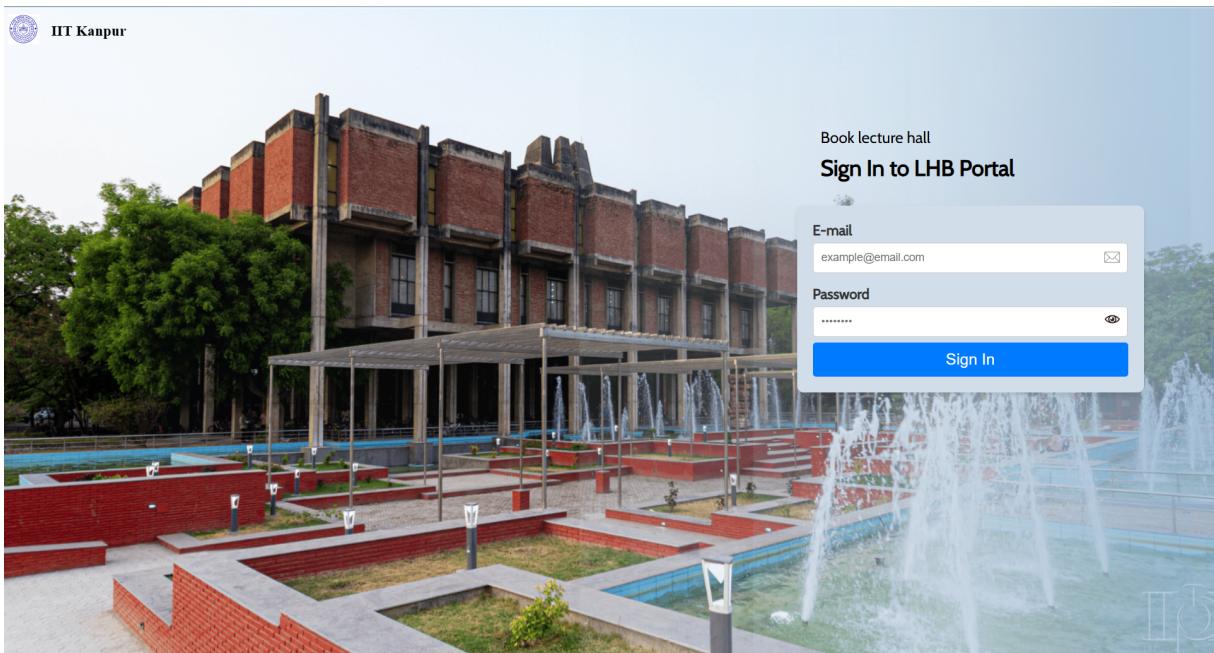


Figure 2: Login Page

## Live Schedule

### view\_timetable(date)

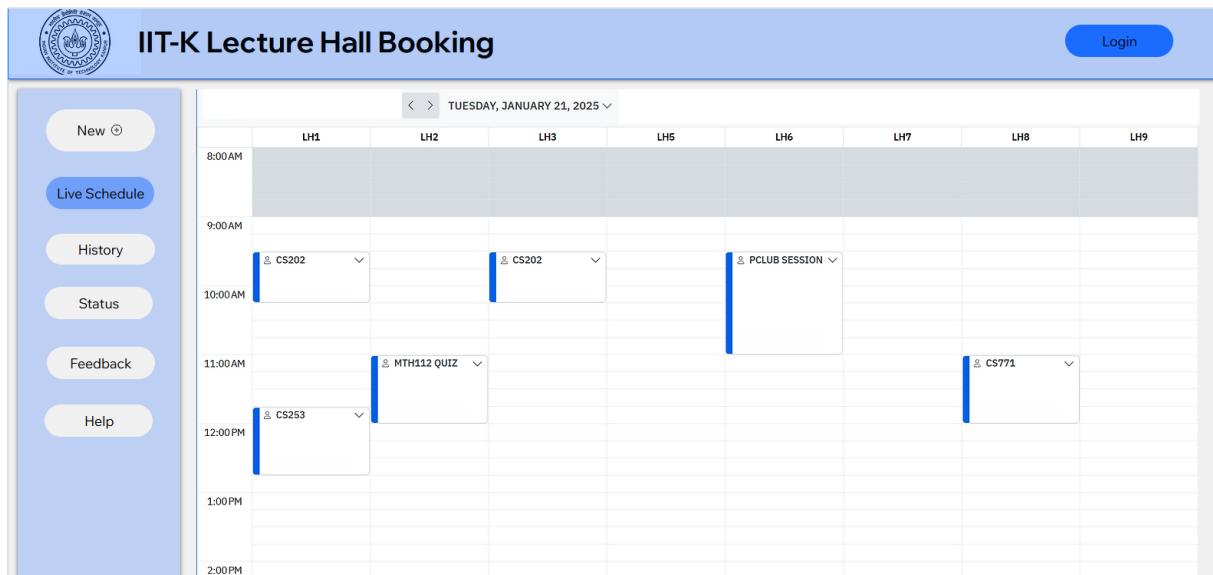


Figure 3: Live Schedule

<b>Request Booking</b>
<code>book_vars = [purpose, user_name, user_type, date_time, repeat, hall, accessories, capacity]</code>
<code>is_admin = True;</code> <code>create_booking(book_vars)</code>
<code>is_admin = False;</code> <code>request_booking(book_vars)</code>

Figure 4: Request Booking

<b>View Booking (User)</b>
<b>view_booking_user(month,year,user_id)</b>
<b>view_bill(booking_id, user_id)</b>

The screenshot shows the 'IIT-K Lecture Hall Booking' application interface. On the left, a sidebar contains buttons for 'New', 'Live Schedule', 'History' (which is selected), 'Status', 'Feedback', and 'Help'. The main area displays a table titled 'Bookings for the Month of January 2025' with columns: Purpose, Date, Time, Lecture Hall, and Amount. The data shows three bookings:

Purpose	Date	Time	Lecture Hall	Amount
Intro Session ML	12-01-25	9:00 AM - 11:00 AM	LH07	6000
Contest	03-01-25	5:00 PM - 7:00 PM	LH03	6000
Workshop Wed-dev	07-01-25	6:00 PM - 7:00 PM	LH07	3500

A total amount of 15500 is displayed at the bottom right. At the top right, there is a 'Logout' button.

Figure 5: User View Booking

<b>View History (Admin)</b>
<b>view_history_admin(month,date)</b>

The screenshot shows the 'IIT-K Lecture Hall Booking' application interface. On the left, a sidebar contains buttons for 'New', 'Live Schedule', 'History' (selected), 'View Pending', 'Create New User', and 'Logout'. The main area shows a booking history for 'TUESDAY, JANUARY 21, 2025' from '9:30AM-11:00 AM (1h 30m)' to '11:45AM-12:45PM (1h)'. A specific booking for 'PCLUB SESSION' is highlighted in blue. A context menu for this booking includes 'Manage', 'View/edit details', and 'Remove booking'. Below this, a list of bookings is shown with columns: Start Time, End Time, Duration, Lecture Hall, Professor, and Room Number. The data includes:

Start Time	End Time	Duration	Lecture Hall	Professor	Room Number
9:30 AM	10:15 AM	45m	LH...	(Prof A)	CS202
11:00 AM	12:00 PM	1h	LH8	(Prof B)	CS771
11:00 AM	12:00 PM	1h	LH2	(Prof C)	MTH112 QUIZ
11:45 AM	12:45 PM	1h	LH1	(Prof D)	CS253

Figure 6: Admin View Booking

View Booking (Admin)		
<code>view_booking(booking_id)</code>		
 <b>IIT-K Lecture Hall Booking</b> <span style="float: right;">Logout</span>		
<span>New </span> <span>Live Schedule</span> <span>History</span> <span>View Pending</span> <span>Create New User</span>	<span style="border: 1px solid blue; padding: 2px;"><b>&lt;</b></span> <b>ECO111</b> <span style="border: 1px solid blue; border-radius: 15px; padding: 2px 10px; margin-left: 10px;">Edit Booking</span> <span style="border: 1px solid blue; border-radius: 15px; padding: 2px 10px; margin-left: 10px;">Delete Booking</span> <b>Description:</b> <b>Approval status:</b> Approved <b>Room:</b> LHC - L 18 <b>Start time:</b> 12:00:00PM - Wednesday 05 February 2025 <b>Duration:</b> 1 hours <b>End time:</b> 01:00:00PM - Wednesday 05 February 2025 <b>Type:</b> Academic <b>Created by:</b> Ihc <b>Last updated:</b> 10:15:03PM - Saturday 23 November 2024 <b>audio_video:</b> YES <b>modified_by:</b> <b>Repeat type:</b> Weekly <b>Repeat day:</b> Monday Tuesday Wednesday Thursday <b>Repeat end date:</b> Wednesday 23 April 2025	

Figure 7: Admin View Booking

Create User (Admin)	
<code>user_details = [user_id, password, email, user_type, verifying_auth, remarks]</code> <code>create_user(user_details)</code>	
 <b>IIT-K Lecture Hall Booking</b> <span style="float: right;">Logout</span>	
<span>New </span> <span>Live Schedule</span> <span>History</span> <span>View Pending</span> <span>Create New User</span>	<span style="border: 1px solid blue; border-radius: 15px; padding: 2px 10px;">User </span> <b>Create New User</b> User ID : <input type="text"/> Password : <input type="password"/> E-mail : <input type="text"/> Category * <input type="radio"/> Academic <input type="radio"/> Non-Academic Verifying Authority * <input type="radio"/> DOSA <input type="radio"/> DOAA <input type="radio"/> General Sec(SnT) <input type="radio"/> None <span style="margin-left: 10px;">Add </span> <span style="float: right;">Confirm</span>

Figure 8: Create User

## 2 Architecture Design

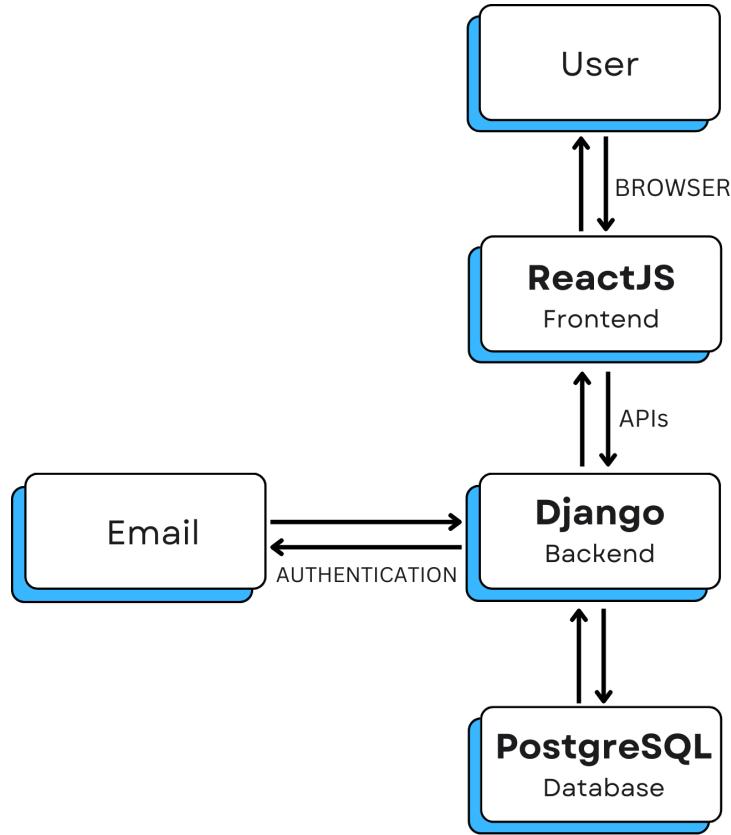


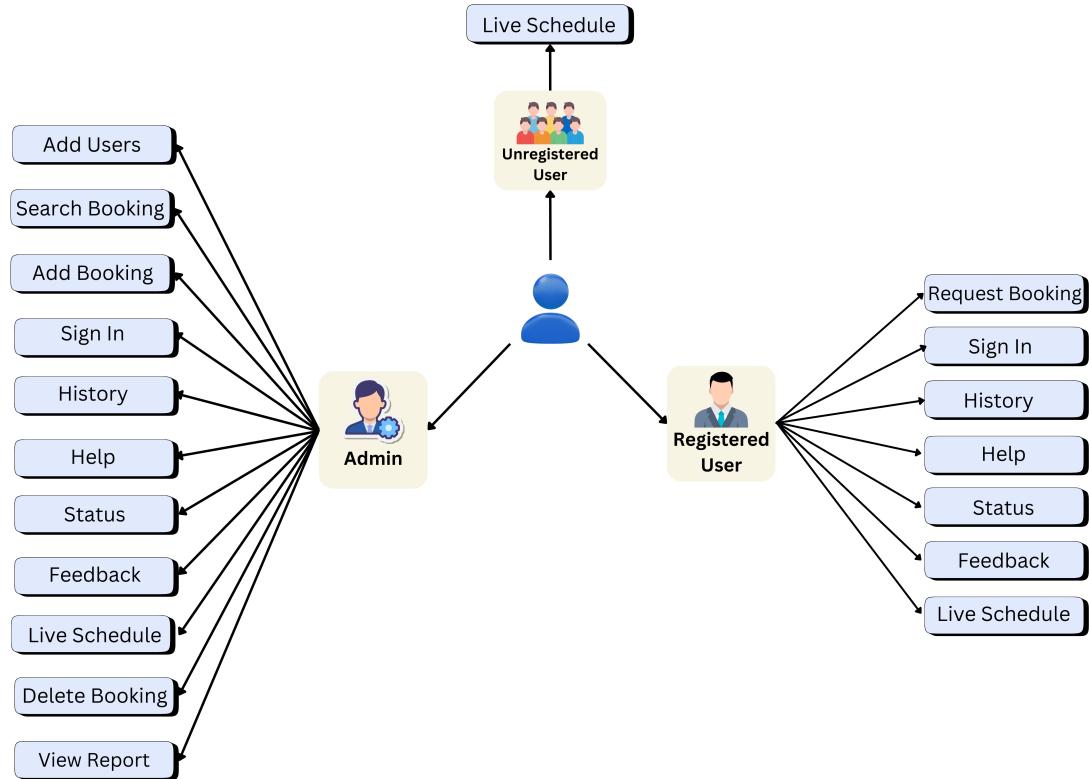
Figure 9: Architecture Flowchart

This architecture should meet the non-functional requirements as described in the SRS document as follows:

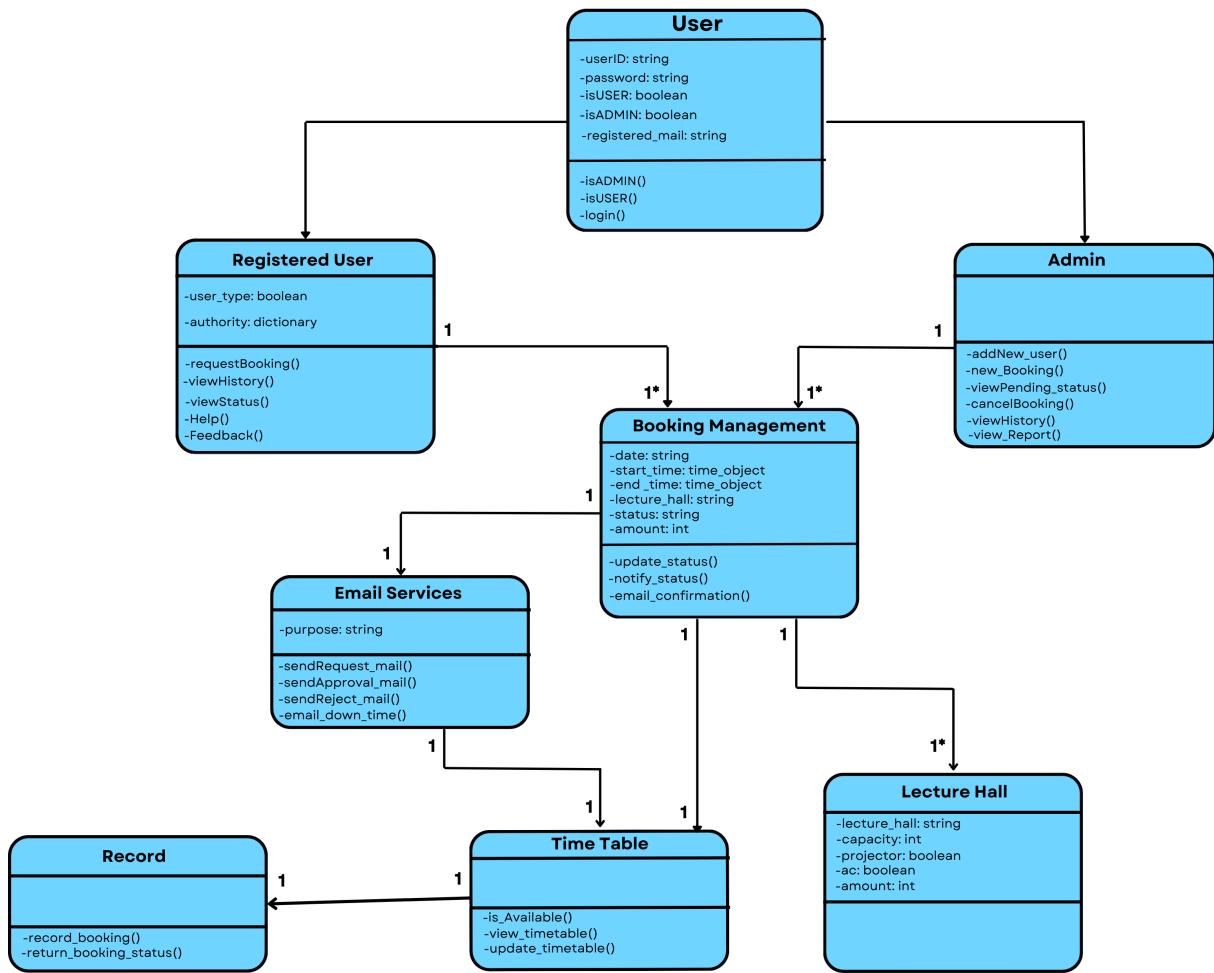
- **Performance Requirements**
  - **Response Time:** All user interface interactions, such as clicking buttons or navigating menus, must respond within 5 seconds to ensure a smooth user experience.
  - **Scalability:** The system must handle a significant increase in users without requiring significant architectural changes, ensuring long-term usability.
- **Safety and Security Requirements**
  - **Authentication and Session Management:** Users must authenticate with unique credentials, and sessions should automatically timeout after 15 minutes of inactivity to ensure security.
- **Software Quality Attributes**
  - **Reliability:** The system must be operational during working hours (8:00 AM to 10:00 PM), with any maintenance scheduled outside this timeframe and communicated to users in advance.

### 3 Object Oriented Design

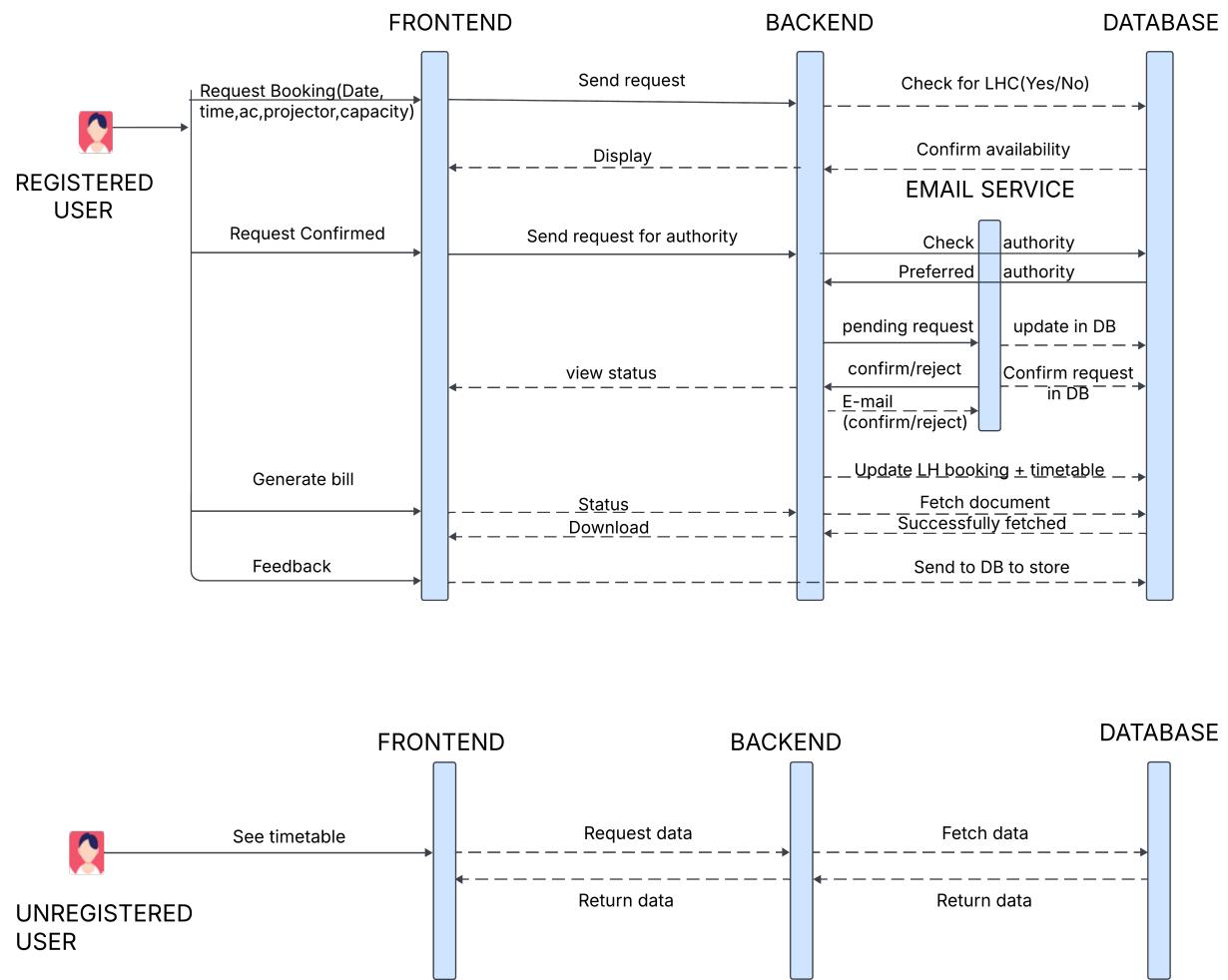
#### 3.1 Use Case Diagram

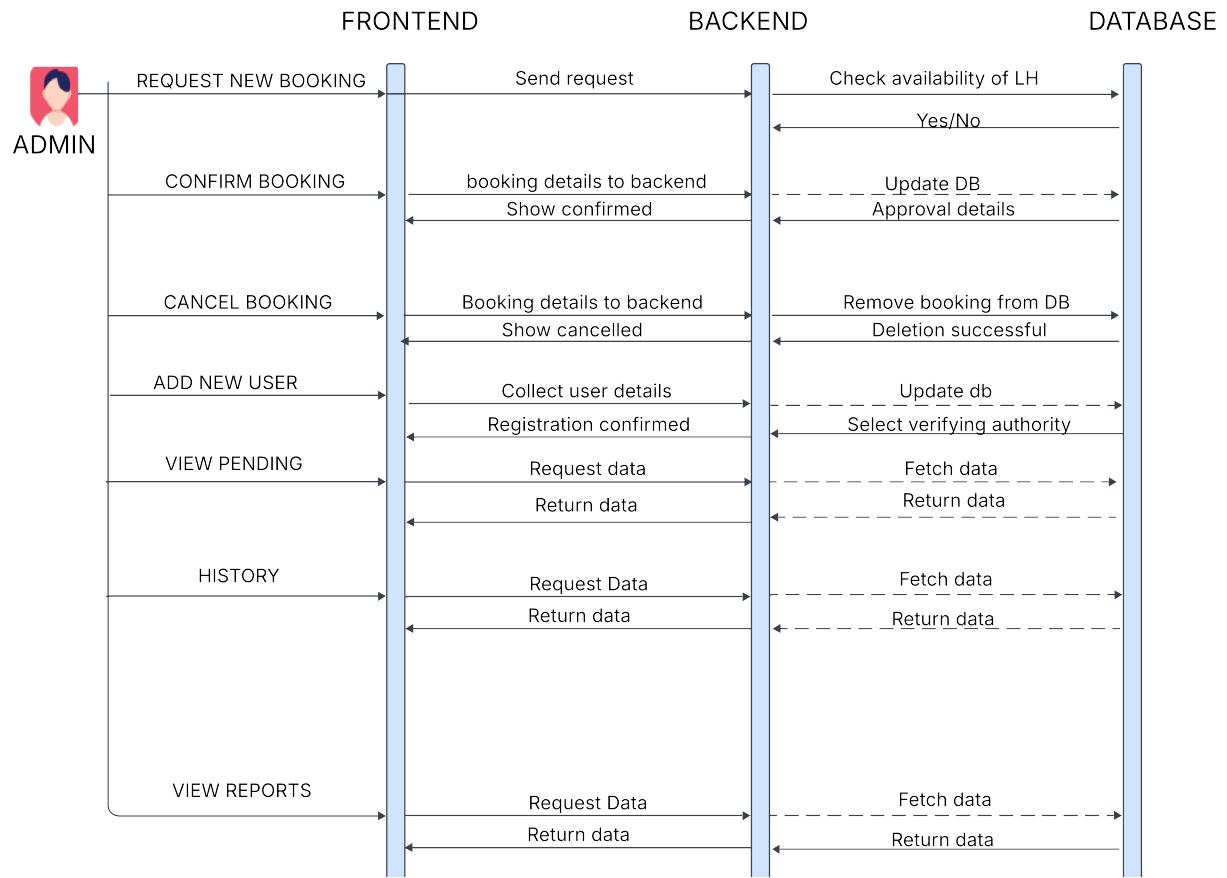


## 3.2 Class Diagram

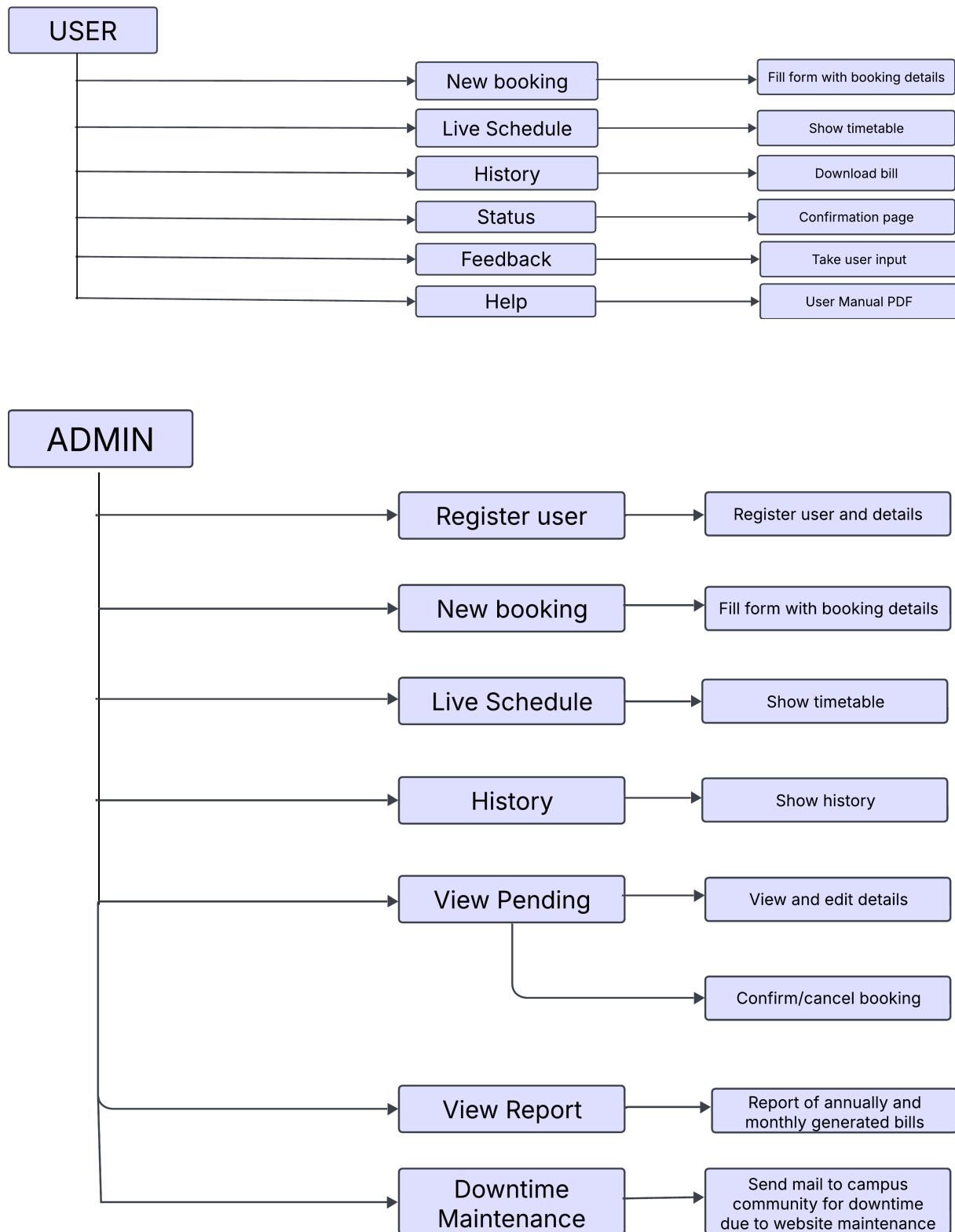


### 3.3 Sequence Diagram

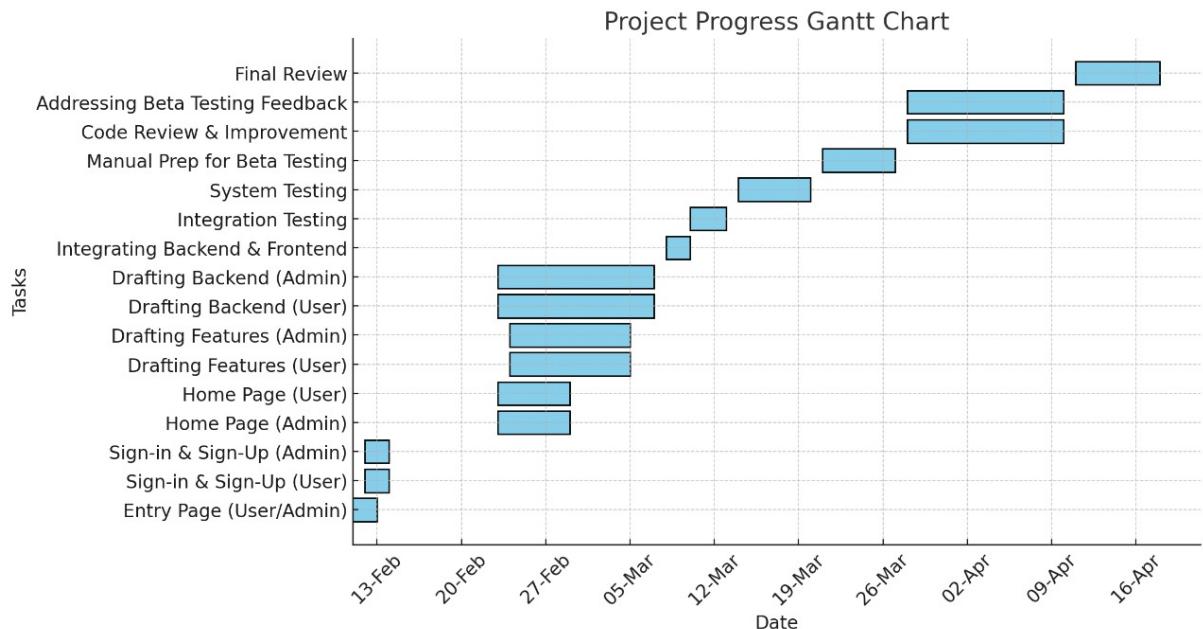




### 3.4 State Diagram



## 4 Project Plan



Activity	Start Date	Duration (days)	End Date	Team Members
Front end: Entry Page (choosing if user or admin)	2024-02-11	3	2024-02-13	Rahul ,Daksh
Front end: Sign-in and Sign-Up page (user)	2024-02-12	3	2024-02-14	Chaitanya ,Atharv
Front end: Sign-in and Sign-Up page (admin)	2024-02-12	3	2024-02-14	Aaradhyा ,Avantika
Front end: Home page (admin)	2024-02-23	7	2024-02-29	Bhavya ,Areen
Front end: Home page (user)	2024-02-23	7	2024-02-29	Devansh ,Divyesh
Front end: Drafting all features for the user's side	2024-02-24	11	2024-03-05	Devansh, Daksh,Divyesh
Front end: Drafting all features for the admin's side	2024-02-24	11	2024-03-05	Aaradhyा ,Avantika,
Backend: Drafting all features for the user's side	2024-02-23	14	2024-03-07	Atharv ,Bhavya ,Areen

<b>Activity</b>	<b>Start Date</b>	<b>Duration</b>	<b>End Date</b>	<b>Team Members</b>
Backend: Drafting all features for the admin's side	23-02-2024	14	07-03-2024	Chaitanya ,Rahul
Integrating the backend and frontend	08-03-2024	3	10-03-2024	Entire Team
Integration Testing	10-03-2024	4	13-03-2024	Entire Team
System Testing	14-03-2024	7	20-03-2024	Entire Team
Manual Preparation for Beta Testing	21-03-2024	7	27-03-2024	Entire Team
Code Review and Improvement	28-03-2024	14	10-04-2024	Entire Team
Addressing Beta Testing Feedback	28-03-2024	14	10-04-2024	Entire Team
Final Review of Project	11-04-2024	8	18-04-2024	Entire Team

## **Appendix A: Group Log**

<b>Meeting minutes</b>	<b>Agenda</b>
25 Jan	Discussed context design (comprising context model and human interface design) and architecture design.
27 - 28 Jan	Focused on object-oriented design, specifically the use case diagram and class diagram.
30 - 31 Jan	Worked on object-oriented design, including the creation of sequence diagrams and state diagrams.
1 Feb	Developed the project plan and distributed tasks among team members.
3 Feb	Compiled all content and integrated it into the LaTeX document for uniformity and structure.
5 Feb	Conducted a final review of all documents to ensure accuracy and completeness.